JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 22, No. 1 : 190~195 Jan. 2018

선택적인 암호화 기능을 지원하는 TCP의 설계 및 구현

성정기 · 김은기*

Design and Implementation of TCP Supporting Optional Encryption Functionalities

Jeong-Gi Seong · Eun-Gi Kim*

Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

요 약

최근 지속되는 사이버 공격의 증가와 개인정보 보호에 대한 인식 강화로 많은 인터넷 서비스는 보안 프로토콜을 사용하여 트래픽을 암호화한다. 기존의 보안 프로토콜은 보통 전송 계층과 응용 계층 사이에서 추가적인 계층을 가지며 전송하는 모든 트래픽을 암호화하므로 추가적인 비용이 발생한다. 이로 인해 기밀성이 요구되지 않는 데이터도 암호화하므로 불필요한 성능저하가 발생한다. 따라서 본 논문에서는 응용 계층의 사용자가 기밀성이 요구되는 데이터만을 선택적으로 암호화할 수 있게 지원하는 TCP OENC(Optional Encryption)를 제안한다 . TCP OENC는 TCP 옵션으로 동작하여 응용 계층이 요구 할 때만 전송되는 TCP 스트림을 암호화하도록 지원하고, TCP 계층과 응용 계층 간의 투명성을 보장한다. 이를 확인하기 위해 구현된 TCP OENC를 개발 보드에서 TCP 세션의 스트림을 선택적으로 암호화하는 것을 검증하였고, 암호화된 스트림의 전송 수행 시간을 측정하여 성능을 분석하였다.

ABSTRACT

Recently, Due to the ongoing increase in cyber attacks and the improved awareness of privacy protection, most Internet services encrypt the traffic by using security protocols. Existing security protocols usually have additional layer between transport layer and application layer, and they incur additional costs because of encrypting all the traffic transmitted. This results in unnecessary performance degradation because it also encrypts data that does not require confidentiality. In this paper, we propose TCP OENC(Optional Encryption) which enables users of the application layer to optionally encrypt only confidential data. TCP OENC operates by TCP option to allow the application layer to encrypt the TCP stream transmitted only on demand. And it ensures transparency between the TCP layer and the application layer. To verify this, we verified that TCP OENC optionally encrypts the stream of TCP session on the embedded board. And then analyzed the performance of the encrypted stream by measuring the elapsed time.

키워드 : TCP, 전송 보안, 암호화, 전송 계층, 네트워크

Key word : TCP, Transport Security, Encryption, Transport Layer, Network

Received 08 November 2017, **Revised** 15 November 2017, **Accepted** 29 November 2017

* Corresponding Author Eun-Gi Kim(E-mail:idreamed64@gmail.com, Tel:+82-42-821-1215)
Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

Open Access http://doi.org/10.6109/jkiice.2018.22.1.190 print ISSN: 2234-4772 online ISSN: 2288-4165

Ⅰ. 서 론

지속되는 사이버 공격의 증가와 개인정보 보호에 대한 인식 강화로 인해 많은 인터넷 서비스들은 트래픽을 암호화하고 있으며, 매년 암호화된 트래픽은 급증하고 있다[1,2]. 인터넷 웹, 파일 전송, 영상 스트리밍과 같은 응용 서비스들은 트래픽을 암호화하기 위해 SSH, SSL/TLS와 같은 보안 프로토콜을 주로 사용한다. 하지만 대부분의 보안 프로토콜은 전송 계층인 TCP 와 응용 계층 사이에 추가적인 계층을 위치하여 동작하므로 성능 측면에서 많은 비용을 발생시키고, 세션에서 발생하는 모든 트래픽을 암호화한다[3,4]. 따라서 응용 계층에서 작은 크기의 데이터만 암호화하는 경우에도 추가적인 보안 프로토콜이 요구되며 또한 기밀성의 구분 없이 모든 데이터를 암호화하므로 불필요한 성능 저하를 발생시키기도 한다. 이와 같은 문제를 보완하고자 본 논문은 TCP에서 추가적인 보안 프로토콜 없이 선택적인 암호화를 지원하는 TCP를 제안한다. 제안하는 TCP은 응용 계층의 사용자가 별도의 보안 프로토콜 없이 데이터를 암호화하는 것을 TCP 옵션으로 지원한다[5]. 또한 응용 계층에서 기밀성이 요구되는 데이터만 선택적으로 암호화할 수 있도록 지원하여 암호화로 발생하는 비용을 줄일 것으로 기대한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 TCP의 설계 및 구현, 3장에서는 동작 검증 및 성능 분석, 4장에서는 결론을 설명한다.

Ⅱ. 설계 및 구현

본 논문에서 제안하는 선택적인 암호화를 지원하는 TCP는 TCP OENC(Optional Encryption)로 기술한다. TCP OENC는 기존의 TCP 수준 암호화 기법과 다르게 TCP 세션에서 사용자의 요구에 따라 전송 데이터를 암호화하도록 지원한다. 그림 1은 TCP OENC의 동작을 나타낸다.

그림 1에 나타나듯이 TCP OENC는 응용 계층에서 암호화를 요구할 때만 전송 데이터를 암호화한다. 최초 암호화 수행 전에 키 협의를 한 번만 수행하고 이후 선택적으로 전송 데이터를 암호화하도록 지원한다. 본 논문에서는 TCP OENC 동작을 수행하도록 리눅스 커널의 TCP를 수정하였고, CryptoAPI를 사용하여 보안 알고리즘을 적용하였다[6].

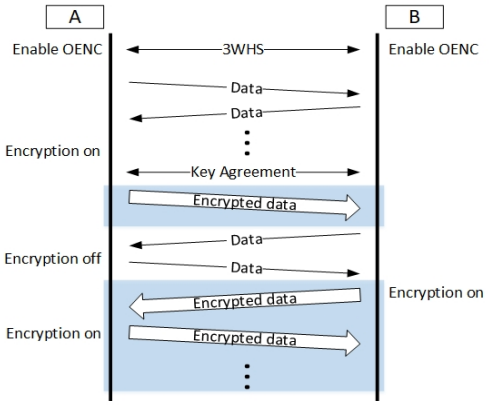

Fig. 1 Overview of TCP OENC

2.1. TCP OENC 옵션

TCP OENC는 기존 TCP와의 하위 호환성을 가지기 위해 TCP 옵션으로 동작한다[7]. 본 논문에서는 TCP 옵션의 KIND를 79로 할당하고 1바이트 크기의 서브 옵션으로 동작을 구분한다. 표 1은 TCP OENC의 서브 옵션을 나타낸다.

Table. 1 TCP ONEC Sub-options

Sub Option	Size	Description
PROBE (1)	5	Probe OENC
PERMIT (2)	5	Permit OENC
INIT (3)	3	Use for key agreement
ENC_ON (4)	7	Notice starting encryption
ENC_OFF (5)	7	Notice stopping encryption

2.2. TCP OENC 옵션 확인 및 보안 알고리즘 협의

TCP OENC는 3WHS를 통해 서로의 OENC 지원여부를 확인하고 보안 알고리즘을 협의한다. TCP OENC는 ECDH(Elliptic Curve Diffie-Hellman)[8,9]와 AES (Advanced Encryption Standard)[10]을 지원한다. 그림 2는 TCP에서 OENC를 사용하는 TCP 3WHS (3-Way Handshaking) 과정을 나타낸다. 3WHS 과정에서 A는 PROBE 옵션을 포함한 SYN를 전송하여 TCP 연결을 요청한다. PROBE 옵션은 상대 호스트의 OENC 지원여부를 확인함과 동시에 자신이 사용 가능한 ECDH와 AES 타입(Type)을 명시한다.

ECDH_LIST와 AES_LIST는 각 1바이트의 크기를 가지며, 각 비트는 알고리즘의 타입을 명시한다. 표 2는 ECDH 타입, 표 3은 AES의 타입을 나타낸다.

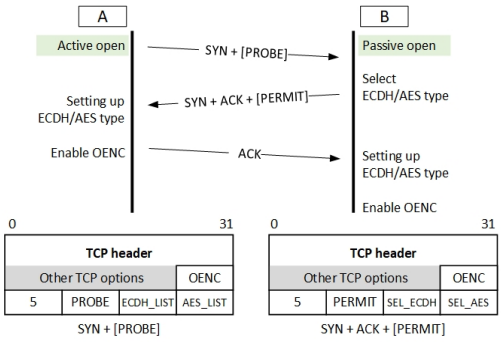

Fig. 2 TCP 3-Way Handshaking with OENC options

Table. 2 ECDH Types of TCP OENC

Bit No.	ECDH Type
0	ECDHE-NIST-P192-SHA512
1	ECDHE-NIST-P256-SHA512

Table. 3 AES Types of TCP ONEC

Bit No.	AES Type
0	AES-128-CTR
1	AES-192-CTR
2	AES-256-CTR

B는 PROBE가 포함된 SYN를 수신하면 현재 세션에서 사용할 ECDH와 AES의 타입을 선택한다. 선택된 ECDH 타입은 SEL_ECDH, AES 타입은 SEL_AES으로 명시한 PERMIT 옵션을 SYN+ACK와 함께 전송한다. A는 PERMIT를 수신하면 선택된 ECDH와 AES 타입으로 보안 알고리즘을 설정하고 OENC 기능을 활성화한 후 ACK를 전송하여 TCP 연결 설정을 완료한다. B는 A의 ACK를 수신하면 OENC 기능을 활성화한 후 연결 설정을 완료한다. 만약 B가 OENC를 지원하지 않는다면 PROBE 옵션은 무시된다[11].

2.3. 키 협의

OENC 기능이 활성화되면 두 호스트는 세션을 유지하는 동안 응용 계층에서 setsockopt() 시스템 콜을 호출하여 자신이 원할 때 데이터를 암호화하여 전송할 수 있다. 하지만 암호화를 수행하기 위해서는 두 호스트만 알 수 있는 비밀키가 필요하다. 두 호스트는 암호화된 데이터 전송 전에 키 협의 동작을 통해 비밀키를 생성한다. 그림 3은 A가 setsockopt()을 먼저 호출한 경우 수행되는 키 협의 과정을 나타낸다.

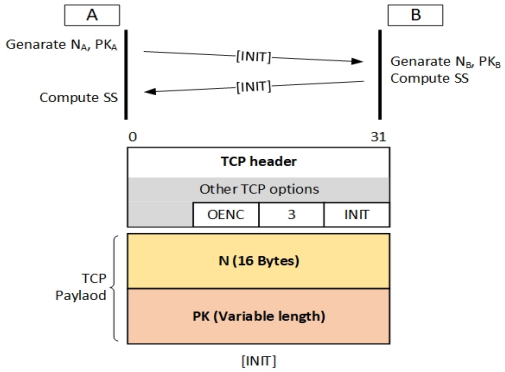

Fig. 3 Key Agreement of TCP OENC

A는 키 협의를 위한 16바이트 길이의 넌스(Nonce) N_A과 ECDH 공개키 PK_A를 생성한다. N_A와 PK_A가 생성되면 INIT 옵션을 통해 B에게 전송한다. 그림 3에서 N_A와 PK_A가 TCP 페이로드로 전송되는데, N_A와 PK_A가 TCP 옵션의 최대 크기 40바이트를 초과하기 때문에 TCP 페이로드를 사용하여 전송한다. 따라서 INIT 옵션이 포함된 세그먼트는 응용 데이터를 포함하지 않으며, 응용 계층으로도 전달되지 않는다.

B가 INIT를 수신하면 N_B와 PK_B를 생성하고 A에게 INIT 옵션과 함께 전송한다. B는 A의 PK_A와 자신의 PK_B를 사용하여 비밀 키 SS(Shared Secret)를 생성한다. 이후 SS와 N_A, N_B을 SHA-512 알고리즘의 입력으로 AES 키와 IV(Initialization Vector)를 생성한다. A도 INIT을 수신하면 B와 동일한 동작을 수행하여 B와 동일한 AES 키와 IV를 생성한다.

2.4. 암호화/복호화 제어

키 협의를 완료하면 두 호스트는 setsockopt()를 사용하여 데이터를 암호화하여 전송할 수 있다. 하나의 호스트가 암호화 동작을 수행하면 상대 호스트는 복호화 동작을 수행해야한다. TCP OENC에서 상대 호스트의 복호화 동작을 제어하기 위해 ENC_ON과 ENC_OFF 옵션을 사용한다. 그림 4는 데이터 암호화/복호화 동작을 처리하는 과정을 나타낸다.

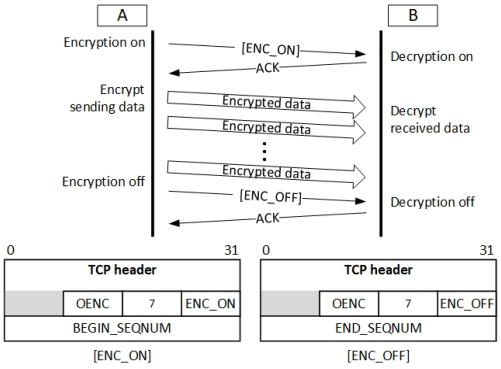

Fig. 4 Encryption/Decryption Control of TCP OENC

그림 4와 같이 A는 데이터를 암호화하기 전에 ENC_ON을 전송한다. ENC_ON은 데이터 암호화 시작을 알리는 옵션으로, 처음 암호화 될 세그먼트의 순서번호 BEGIN_SEQNUM을 포함한다. B는 ENC_ON을 수신하면 BEGIN_SEQNUM의 세그먼트부터 데이터를 복호화하여 응용 계층에 전달한다. 만약 A가 데이터 암호화를 중지할 경우, 암호화 중지를 알리는 ENC_OFF를 전송한다. ENC_OFF의 END_SEQNUM은 암호화되지 않은 첫 번째 세그먼트의 번호이며 ENC_OFF을 수신한 B는 END_SEQNUM의 세그먼트부터 데이터를 복호화하지 않는다.

Ⅲ. 동작 검증 및 성능 분석

본 논문에서 제안한 TCP OENC의 동작을 확인하고 성능을 분석하기 위해 리눅스 커널 4.12.14의 TCP을 수정하여 개발 보드[12]에서 동작시켰다. TCP 통신을 위한 네트워크 환경은 유선 공유기[13]를 이용하여 구성하였고, 포트 미러링 기능과 와이어샤크 S/W로 개발 보드의 TCP 스트림을 확인하였다. 표 4와 그림 5는 측정에 사용된 개발 보드와 LAN 환경을 나타낸다.

Table. 4 Embedded Board Specification for Test

Model	FALinux EZ-S3C6410
MCU	667MHz S3C6410 ARM RISC Chip
RAM	Mobile DDR 128 MB (64MB x 2)
Ethernet	10/100Base-T
OS	Linux kernel 4.12.14

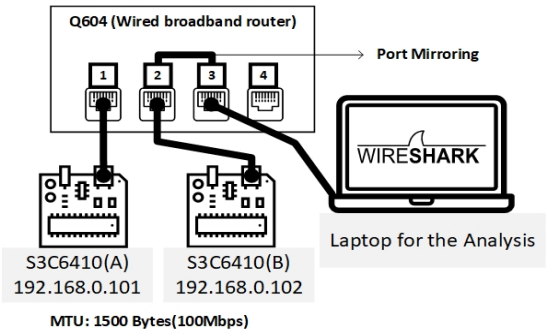

Fig. 5 LAN Environment for the Test

3.1. TCP OENC 동작 검증

본 논문에서는 TCP OENC가 하나의 TCP 세션에서 데이터 스트림을 선택적으로 암호화하는 동작하는 것을 검증하기 위해 HTTP 클라이언트/서버 모델의 응용 프로그램을 구현하였고, ECDHE-NIST-P256-SHA512와 AES-256-CTR 알고리즘을 사용하였다.

동작 검증에서는 HTTP 클라이언트가 512 바이트 크기의 텍스트 파일(512B.txt)을 HTTP 요청하고 HTTP 응답으로 파일을 전송하는 동작을 와이어샤크로 확인하였다. 암호화 여부를 쉽게 확인하기 위해 'A'부터 'Z'까지의 연속적인 알파벳 문자들로 구성된 512B.txt 텍스트 파일을 2번째와 4번째 부분만 암호화하여 전송하였다. 그림 6은 TCP 연결 설정에서 HTTP 클라이언트 A가 전송하는 SYN과 HTTP 서버 B가 전송하는 SYN+ACK를 캡처한 화면이다. 그림 6에서 나타나듯이 SYN과 SYN+ACK에 PROBE 옵션과 PERMIT 옵션이 추가되어 동작되는 것을 확인하였다. 그림 7은 B가 전송하는 INIT 패킷을 캡처한 화면이고, 그림 8은 ENC_ON과 ENC_OFF을 캡처한 화면이다.

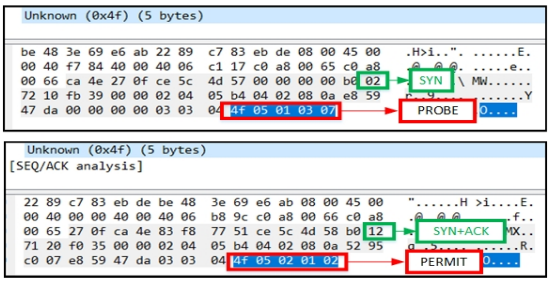

Fig. 6 PROBE and PERMIT Packets Capture

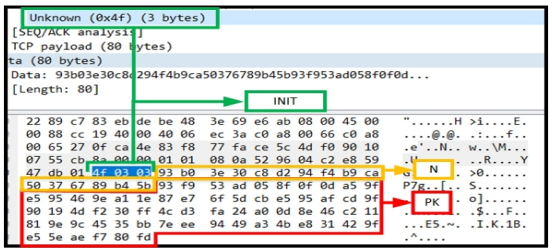

Fig. 7 INIT Packet Capture

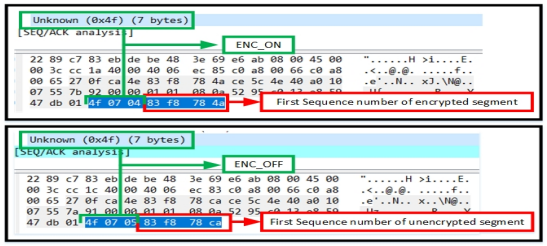

Fig. 8 ENC_ON and ENC_OFF Packet Capture

그림 9는 512B.txt의 HTTP 요청/응답 동작의 TCP 스트림을 와이어샤크로 캡처한 화면이다. 그림 9에 나타나듯이 INIT을 통해 키 협의를 수행하고 2번째와 4번째 스트림의 128바이트만 암호화되는 것을 확인하였다.

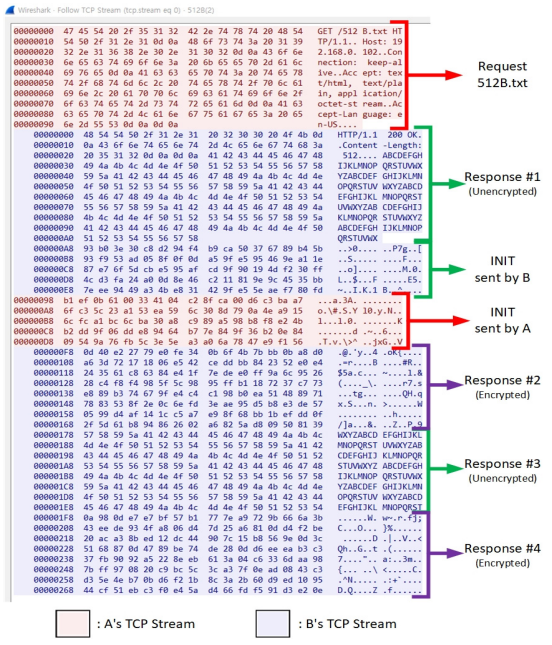

Fig. 9 TCP OENC Stream Capture

3.2. TCP OENC 성능 분석

성능 분석에서는 하나의 TCP 세션에서 암호화를 수행하지 않는 동작을 기준으로 트래픽 절반만 암호화하는 동작, 트래픽 전체를 암호화하는 동작에 대한 성능을 비교하였다. 성능 비교를 위해 10개의 동일한 파일에 대한 HTTP 요청/응답 동작 처리 시간을 측정하였다. 각 성능 측정에서 사용된 파일은 1KB, 10KB, 100KB, 1000KB의 크기를 가지며, 소요 시간 측정은 HTTP 클라이언트에서 10개의 요청/응답 수행이 모두 완료되는 시간을 clock_gettime() 함수를 이용하여 측정하였다. 이와 동일한 동작을 100번 수행하여 평균 시간을 계산하였다. 표 5는 측정 시간의 평균 시간을 나타낸다.

표 5에 나타나듯이 트래픽 절반만 암호화한 동작이 모든 트래픽을 암호화한 동작보다 빠르게 수행되는 것을 확인하였다. 또한 전송되는 트래픽이 클수록 처리 시간이 지연되는 폭이 커지는 것도 확인하였다. 트래픽을 암호화하지 않은 동작을 기준으로 1KB 파일의 경우에는 트래픽 전체를 암호화한 동작이 약 23.67%, 트래픽 절반을 암호화한 동작이 약 17.31% 정도 처리 시간이 지연되었고, 1000KB 파일의 경우에는 트래픽 전체를 암호화한 동작이 약 41.24%, 트래픽 절반만 암호화한 동작이 약 24.79% 로 암호화로 인한 처리 시간 지연이 발생한 것을 확인하였다.

Table. 5 The Average Elapsed Times

File Size	Unencrypt	Encrypt half traffic	Encrypt entire traffic
1KB	29.13 μs	35.23 μs	38.17 μs
10KB	73.53 μs	91.38 μs	108.0 μs
100KB	430.18 μs	567.73 μs	720.71 μs
1000KB	3963.4 μs	5269.43 μs	6745.31 μs

Ⅳ. 결 론

본 논문에서는 기존의 트래픽을 암호화하는 과정에서 모든 데이터를 암호화하여 기밀성이 요구되지 않는 경우에도 발생하는 불필요한 성능 저하를 줄이고자 TCP에서 선택적인 암호화를 지원하는 TCP OENC를 설계 및 구현에 관해 기술하였다.

본 논문에서 제안한 TCP OENC는 TCP 옵션으로 동작하여 기존의 TCP와 하위 호환성을 가지며 추가적인

보안 프로토콜 없이 데이터를 암호화하는 것을 지원한다. 응용 계층에서 setsockopt() 호출로 TCP 트래픽의 암호화 기능을 설정하여 기밀성이 필요한 데이터를 암호화할 수 있고, TCP가 복호화 동작에 대해 독립적으로 수행하여 응용 계층에 대한 투명성을 보장한다. 추가적으로 동작 검증을 통해 TCP OENC가 추가적인 보안 프로토콜 없이 TCP 데이터 스트림을 부분적으로 암호화하는 것을 확인하였다. 이 결과로 응용 계층에서 기밀성이 필요한 데이터만 암호화할 수 있고, 전송 스트림을 선택적으로 암호화하는 것이 성능 저하를 감소시키는 것으로 확인하였다. 결과적으로 기밀성이 필요한 데이터에 대해서만 암호화를 수행한다면 불필요한 성능 저하를 줄일 수 있을 것으로 기대된다.

REFERENCES

[1] Betanews. Increase network traffic encryption ... 'SSL/TLS decryption·inspection' requires strategic approach [Internet]. Available: http://www.betanews.net/article/626452.

[2] Boannews. SSL-encrypted traffic utilization is expected to increase 10% in 2017 [Internet]. Available: http://www.boannews.com/media/view.asp?idx=57871&mkind=1&kind=1.

[3] Computer world. Encryption, not necessarily good [Internet]. Available: http://www.comworld.co.kr/news/articleView.html?idxno=5413.

[4] Digicert. How Does the SSL Certificate Create a Secure Connection? [Internet]. Available: https://www.digicert.com/ssl/.

[5] J. G. Seong and E. G. Kim, "A Study on the TCP Supporting Optional Encryption,," in *Proceeding of the 42th Conference of Korea Institute of Information and Communication Engineering*, Cheonan, pp. 565-568, 2017.

[6] The Linux Kernel Archives. Linux Kernel Crypto API [Internet]. Available: https://www.kernel.org/doc/html/v4.12/crypto/intro.html.

[7] Charles M. Kozierok, "TCP Message formatting and Data Transfer," in *The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference*, San Francisco, CA: No Starch Press., ch. 48, pp. 773, 2005.

[8] K. H. Kim, "Comparison and analysis on efficiency of scalar multiplication for Elliptic Curve Cryptosystem," M.S. dissertation, Korea Maritime and Ocean University, Busan, 2003.

[9] S. H. Sun and E. G. Kim, "The automatic generation of MPTCP session keys using ECDH," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 20, no10, pp. 1912-1918, Oct. 2016.

[10] S. M. Kim, T. M. Chang, H. S. Kim, and M. S. Kang, "Design of High-Speed AES Cipher Processor Using Pipeline Technique," *Journal of Security Engineering*, vol. 11, no.2, pp. 145-154, Apr. 2014.

[11] RFC 1122, *Requirements for Internet Hosts -- Communication Layers*, IETF, Fremont, CA., 1989.

[12] FALiNUX Forum. EZ-S3C6410 [Internet]. Available: http://forum.falinux.com/zbxe/index.php?mid=EZS3C6410.

[13] EFM-ipTime. Product | EFM - ipTime Q604 [Internet]. Available: http://iptime.com/iptime/?page_id=11&pf=15&page=&pt=114&pd=3.

성정기(Jeong-Gi Seong)

2016년 2월: 한밭대학교 정보통신공학과 (공학사)
2016년 3월~현재: 한밭대학교 정보통신전문대학원 석사 과정
※관심분야 : 네트워크 보안, 임베디드 S/W, 컴퓨터 네트워크

김은기(Eun-Gi Kim)

1989년 2월: 고려대학교 대학원 전자공학과 (전자공학 석사)
1994년 2월: 고려대학교 대학원 전자공학과 (전자공학 박사)
1995년 3월~현재: 한밭대학교 정보통신공학과 교수
※관심분야 : 컴퓨터 네트워크, 임베디드 S/W, 암호화, 네트워크 보안